

---

# Opérateurs de l'algèbre relationnelle

- Opérateurs ensemblistes
- Produit cartésien

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Opérateurs ensemblistes

---

- L'UNION: fusionne des données de deux tables (opérateurs **UNION** et **UNION ALL**)
- L'intersection: extrait des données présentes simultanément dans les deux tables (opérateur **INTERSECT**)
- La différence: extrait des données présentes dans une table sans être présentes dans la 2<sup>ème</sup> table (opérateur **MINUS**).

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Opérateurs ensemblistes

## Syntaxe:

```
SELECT    ...
FROM     table
[WHERE...] [GROUP BY...]
UNION | INTERSECT | MINUS
SELECT...
FROM     table
[WHERE...] [GROUP BY...] ;
```

Il faut que les expressions dans les listes **SELECT** comportent le même nombre de colonnes, dans le même ordre et aussi le même type de données.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Opérateurs ensemblistes

## Exemple :

Pour connaître les villes qui sont soit des villes de départ d'un vol, soit des villes d'arrivée d'un vol.

```
SELECT VilleD Ville
FROM Vol
WHERE VilleD is not null
UNION
SELECT VilleA
FROM Vol
WHERE VilleA is not null;
```

VILLED	VILLEA	RESULTAT
MARSEILLE	MARSEILLE	AJACCIO
MOSCOU	MOSCOU	ALGER
AMSTERDAM	AMSTERDAM	AMSTERDAM
NANTES	PEKIN	BEYROUTH
BEYROUTH	BEYROUTH	ISTANBUL
ALGER	ALGER	MARSEILLE
PARIS	PARIS	MOSCOU
AJACCIO	ISTANBUL	NANTES
NICE	NEW YORK	NEW YORK
	NICE	NICE
		PARIS
		PEKIN

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Produit cartésien

- **PRODUIT CARTESIEN**: disposer de 2 tables dans la clause FROM ce qui permet de composer des combinaisons à partir des données des deux tables.

```
SELECT ...
FROM table[alias],...,table2 [alias]
```

ou

```
SELECT ...
FROM table1 CROSS JOIN table2
```

- La commande **CROSS JOIN** est un type de jointure sur 2 tables SQL qui permet de retourner le produit cartésien.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Produit cartésien

## Exemple :

Liste de tous les vols des compagnies

```
SELECT *
FROM Compagnie C CROSS JOIN Vol V;
```

COMP	NRUE RUE	VILLE	NOMCOMP	NUMVOL	NUMP	NUMAV	DATEA	DATED	VILLEA
AF	2 blaise	Marseille	Air France	900	2000	727	03/05/03	04/05/03	marseille
AC	2 blaise	Marseille	Air Compagnie	900	2000	727	03/05/03	04/05/03	marseille
SING	2 blaise	Singapour	Air Singapour	900	2000	727	03/05/03	04/05/03	marseille
AF	2 blaise	Marseille	Air France	1900	2001	737	03/05/18	04/05/18	Paris

---

# Jointures SQL

Si l'on met une ou plusieurs conditions entre attributs de différentes tables, on ne parle plus de produit cartésien mais de **jointure**

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointures

---

- Permettent d'extraire des données issues de plusieurs tables
- Une jointure met en relation deux tables sur la base d'une clause de jointure (comparaison de colonnes).
- Types de jointure
  - **Jointure naturelle:** NATURAL JOIN
  - **Jointure interne:** INNER JOIN
  - **Jointure externe:** (LEFT ou RIGHT ou FULL) OUTER JOIN
  - **Auto-jointure**

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Conditions de jointure

---

Les conditions de jointures spécifient comment les tables sont jointes (quels attributs et quelle comparaison)

- Équi-jointure = la condition de jointure est une égalité
- Inéqui-jointure = la condition de jointure n'est pas (forcément) une égalité

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure naturelle

- La condition de jointure applique une égalité entre les attributs identiques (même label) des deux tables :
  - S'effectue à la condition qu'il y ai des colonnes du même nom et de même type dans les 2 tables.
- Le résultat d'une jointure naturelle est la création d'un tableau avec autant de lignes qu'il y a de paires correspondant à l'association des colonnes de même nom.

**Syntaxe:** L'opérateur **NATURAL JOIN** permet d'éviter de préciser les colonnes concernées par la jointure.

```
SELECT    liste d'expressions
FROM      table1 NATURAL JOIN table2
[WHERE   conditions]
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure naturelle

## Exemple :

Pour connaître la capacité de chaque vol qui a été affecté d'un avion.

```
SELECT      NumVol, CapAv
FROM        Vol NATURAL JOIN Avion;
```

**Vol**

NUMVOL	NUMAV
101	560
141	101
169	101
631	Null
801	240

**Avion**

NUMAV	Capav
101	350
240	Null
560	250

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure naturelle

Exemple :

VOL	
NUMVOL	NUMAV
101	560
141	101
169	101
631	Null
801	240

AVION

NUMAV	Capav
101	350
240	Null
560	250

Jointure  
naturelle  
sur NumAv

NUMAV	NUMVOL	Capav
560	101	250
101	141	350
101	169	350
240	801	Null

Projection sur NumVol, CapAv

NUMVOL	Capav
101	250
141	350
169	350
801	Null

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

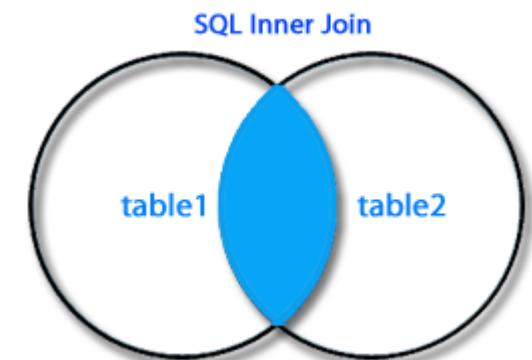
# Jointure interne

La **jointure interne** est fréquemment utilisée : seuls les tuples qui respectent la condition de jointure sont conservés

## Syntaxe:

```
SELECT      TABLE1.COLONNE1, TABLE2.COLONNE2...
FROM        TABLE1 [alias]
            [INNER] JOIN
        TABLE2 [alias]
ON         TABLE1.COLONNE1 comp TABLE2.COLONNE2
```

Avec  $\text{comp} \in \{=, <>, <, \leq, >, \geq\}$



- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure interne

## Exemple :

Pour connaître les numéros de vols des pilotes qui habitent Marseille.

```
SELECT      numvol
FROM        Vol V INNER JOIN Pilote P ON V.NumP=P.NumP
WHERE       VilleP = 'Marseille' ;
```

- Remarque: Utiliser des alias pour les tables pour améliorer les performances

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure interne

Remarque: Différence **Jointure naturelle** et **Jointure interne**

```
SELECT *
FROM Vol NATURAL JOIN Avion;
```

**VOL**

NUMVOL	NUMP	NUMAV	VILLEED	VILLEA	DATED	DATEA
12	200	120	MARSEILLE	PARIS		
13	300	130	Marseille	Nice		

**AVION**

NUMAV	AVNOM	NBVHOL	CAPAV LOC	COMP
120	Boeing	12	30 marseille	SING
130	Sport	13	40 Paris	AF

NUMAV	NUMVOL	NUMP	VILLEED	VILLEA	DATED	DATEA	AVNOM	NBVHOL	CAPAV LOC	COMP
120	12	200	MARSEILLE	PARIS			Boeing	12	30 marseille	SING
130	13	300	Marseille	Nice			Sport	13	40 Paris	AF

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure interne

Remarque: Différence **Jointure naturelle** et **Jointure interne**

```
SELECT *
FROM Vol V INNER JOIN Avion A on
V.NumAv = A.NumAv ;
```

NUMVOL	NUMP	NUMAV	VILLEED	VILLEA	DATED	DATEA
12	200	120	MARSEILLE	PARIS		
13	300	130	Marseille	Nice		

**VOL**

NUMAV	AVNOM	NBHVOL	CAPAV	LOC	COMP
120	Boeing	12	30	marseille	SING
130	Sport	13	40	Paris	AF

**AVION**

NUMVOL	NUMP	NUMAV	VILLEED	VILLEA	DATED	DATEA	NUMAV	AVNOM	NBHVOL	CAPAV	LOC
12	200	120	MARSEILLE	PARIS			120	Boeing	12	30	marseille
13	300	130	Marseille	Nice			130	Sport	13	40	Paris

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure externe gauche ou droite ou complète

C'est une opération entre une table appelée table externe et une autre table appelée table de base (identifiée par LEFT ou RIGHT).

```
SELECT    ...
FROM      table1 [alias]
          {LEFT | RIGHT | FULL} [OUTER] JOIN
          table2 [alias]
          ON exp_col comp exp_col
          [AND | OR exp_col comp exp_col] ...
```

- Le résultat de la jointure externe est la jointure normale des deux tables augmenté des lignes de la table de base qui n'y figurent pas. Pour ces lignes, il y a alors absence de valeur sur les colonnes de la table externe.
- Les jointures externes permettent d'extraire des enregistrements qui ne répondent pas aux critères de jointure.

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure externe

## Exemple :

Pour connaître le nombre de places de chaque vol (même lorsque aucun avion n'est affecté au vol).

```
SELECT  NumVol, CapAv
FROM    Vol V LEFT OUTER JOIN Avion A
        ON V.NumAv = A.NumAv ;
```

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure externe gauche

Exemple :

VOL	NUMVOL	NUMP	NUMAV	AVION
12	200	120		
13	300	130		
15	500	(null)		

NUMAV	AVNOM	NBHVOL	CAPAV
120	Boeing	12	30
130	Sport	(null)	40
200	Test	14	15
400	Boeing	15	20

↓ Equi jointure externe

NUMVOL	NUMP	NUMAV	VILLED	VILLEA	DATED	DATEA	NUMAV	AVNOM	NBHVOL	CAPAV	LOC
12	200	120	MARSEILLE	PARIS			120	Boeing	12	30	marseille
13	300	130	Marseille	Nice			130	Sport		40	Paris
15	500		Paris	Marseille							

Projection sur  
numVol, CapAv

NUMVOL	CAPAV
12	30
13	40
15	

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure externe

## Exemple :

1)

```
SELECT      NumVol, CapAv
FROM        Vol V RIGHT OUTER JOIN Avion A
          ON V.NumAv = A.NumAv ;
```

NUMVOL	CAPAV
12	30
13	40
	15
	20

2)

```
SELECT      NumVol, CapAv
FROM        Vol V INNER JOIN Avion A
          ON V.NumAv = A.NumAv ;
```

NUMVOL	CAPAV
12	30
13	40

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure externe

## Exemple :

3)

```
SELECT      NumVol,   CapAv
FROM    Vol V FULL OUTER JOIN Avion A
        ON V.NumAv = A.NumAv ;
```

NUMVOL	CAPAV
12	30
13	40
15	20
	15

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Jointure externe

## Exemple :

Pour connaître le nombre de vols que chaque pilote assure.

```
SELECT      P.NumP, count(NumVol) AS NbVols
FROM        Pilote P LEFT JOIN Vol V
ON          P.NumP = V.NumP
GROUP BY    P.NumP ;
```

NUMP	NOM	NBVOL	VILLEP	COMP
200	PATRICK	20	MARSEILLE	AF
300	TEST	30	Paris	SING
500	MARC	15	Paris	AF

PILOTE

NUMVOL	NUMP	NUMAV
12	200	120
13	300	130

VOL

## Résultat

NUMP	NBVOLS
500	0
300	1
200	1

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Auto-jointure

---

- Auto-jointure = jointure d'une table avec elle même (tous types de jointures acceptés, i.e., interne, externe), en utilisant des alias de table.
- Exemples fréquents d'auto-jointure : *personne/parents, employé/supérieur hiérarchique, pièce/composant*

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Auto-jointure

## Exemple

Les paires de pilotes qui ont conduit le même avion

```
SELECT DISTINCT V1.numP as P1, V2.numP as P2
FROM Vol V1 INNER JOIN VOL V2 ON V1.numAv =V2.numAv
WHERE V1.numP < V2.numP ;
```

NUMVOL	NUMP	NUMAV	P1	P2
12	200	120		
13	300	130		
15	500	(null)		
25	200	130	200	300

- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# Auto-jointure

## Exemple

Les paires de pilotes qui ont conduit le même avion

V1 . numP	V1 . NUMAV
200	120
300	130
200	130

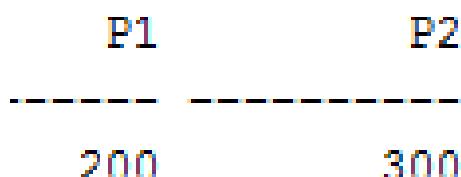
V2 . NUMAV	V2 . numP
120	200
130	300
130	200



V1 . numP	V1 . NUMAV	V2 . numP
200	120	200
300	130	300
200	130	200
300	130	200
200	130	300

Jointure sur  
 $V1.numAv=V2.numAv$

↓  
Restriction sur  
 $V1.numP < V2.numP$



- Sélection
- Recherche
- Ordonner les réponses

- Fonctions de groupe
- Sous-requêtes
- Opérateurs de l'algèbre relationnelle

# En résumé

---

- La condition = naturelle, équi-jointure ,inéqui-jointure
- Les n-uplets conservés dans le résultat= jointure interne, jointures externes (et le produit cartésien)

